Assignment 8

Maggie Schweihs

10/25/2016

1. Pleasantness Score

In this problem, you will create and apply a function that rates cities based on how appealing they are for you to live in.

Create an R function that computes the pleasantness score of a city, based on a vector of data about it. You may assume that the vector contains data in the same order as it is listed in Best Cities.csv.

Added Variables

I started by adding a 5 rows of data to the dataset. This information was gathered from http://factfinder.census.gov/. The following are the added rows to "Best Cities.csv":

- Employment Status: Armed Forces(2014)
- Worked at Home (2014)
- Number of establishments in the Information Industry (2012)
- Number of workers in the information industry (2012)
- Annual payroll for workers in the information industry(in $1000s)(2012)
- InfoPay_Avg is the annual payroll (2014) divided by Number of workers in Information Industry

I chose to add more data regarding people employed in the Armed Forces because I am currently a member of the National Guard. My assumption is that people who report the "Armed Forces" in their employment information are either members of the reserves or active military members. Either way, cities with a higher number of Armed Forces employees should be better suited for someone who is a member of the Armed Forces.

I also added the number of people who reported that they worked at home as a key indicator of a pleasant place to live. I belive this number could indicate the level of work-life balance in certain areas.

Finally, I added several measures of the "Information" Industry. I wanted to know which cities had the most establishments dedicated to the Information Industry, how many people were employed and the annual payroll for those employees. I used Annual payroll and number of workers in Information Industry to calculate the "InfoPay_Avg" column as an estimate of the average salary of people working in the Information Industry in that city.

Data Preparation

I modified the "Best Cities.csv" and created "myBestCities.csv" for the purpose of running the scoring algorithm. First, I added the rows listed above. Second, I transposed the rows and columns so the variables were the columns and the cities were the rows. Thrid, I renamed the columns so they would be easier to call in the functions. Fourth, in excel, I converted all of the cells to "numeric" since I was receiveing type errors such as the following:

Warning message: In Ops.factor(InfoPay, InfoWorkers) : '/' not meaningful for factors

```
colnames(myBestCities)
```

```
##  [1] "Bachelors"            "Commute"              "Rent"
##  [4] "Income"               "OwnerCostsMortgage"   "OwnersCostNoMortgage"
##  [7] "HomeValue"            "OwnOccupiedRate"      "PerCapIncome"
## [10] "NoHealthIns"          "Poverty"              "PopSqMile"
## [13] "Pop"                  "VetOwned"             "Vets"
## [16] "Bike"                 "MaxTemp"              "MinTemp"
## [19] "ArmedForces"          "WorkAtHome"           "InfoEst"
## [22] "InfoWorkers"          "InfoPay"              "InfoPay_Avg"
```

```r
rownames(myBestCities)
```

```
##  [1] "Madison.city..Wisconsin"          "Minneapolis.city..Minnesota"
##  [3] "San.Francisco.city..California"   "Austin.city..Texas"
##  [5] "Philadelphia.city..Pennsylvania" "New.York.city..New.York"
##  [7] "Los.Angeles.city..California"     "Seattle.city..Washington"
##  [9] "Portland.city..Oregon"            "Miami.city..Florida"
## [11] "Charlottesville.city..Virginia"
```

Scoring

I divided my scoring criteria into three main categories: Life Style, Information Industry, and Veteran/Armed Forces Friendly.

Scoring parameters conform to the following naming convention:

- Life Style = L_name
- Information Industry = I_name
- Veteran/Armed Forces Friendly = V_name

Where *name* is a meaningful name for the parameter.

For each parameter, the function takes the index of the city as its only input.

Life Style

- Rent: Lower is better

```r
L_rent <- function(x){ #begin L_rent: inverse of Rent
  return(1/Rent[x]*1000)
} #end L_rent
```

- Commute: Lower is better

```r
L_commute <- function(x){ #begin L_commute: inverse of commute
  return(1/Commute[x]*10)
} #end L_commute
```

- Population/sq. mile: Lower is better

```r
L_popDensity <- function(x){ #begin L_popDensity
  return(1/PopSqMile[x]*1000)
} #end L_popDensity
```

- Proportion of population that works at home: higher is better

```r
L_telework <- function(x){ #begin L_telework
  return(WorkAtHome[x]/Pop[x]*10)
} #end L_telework
```

- Median Value of Owner Occupied Homes: less than $250,000 is best

```
L_homeVal <- function(x){ #begin L_homeVal
  if( HomeValue[x] < 250000){
    return(1)
  }
  else return(0)
} #end L_homeVal
```

Information Industry

- Proportion of workers in information industry: higher is better

```
I_nerdPopDensity <- function(x){ #begin I_nerdPopDensity
  return(InfoWorkers[x]/Pop[x]*10)
} #end I_nerdPopDensity
```

- Average annual salary per employee in information industry: between $65,000 and $100,000 is best.

```
I_nerdPay <- function(x){ #begin I_nerdPay
  if(InfoPay_Avg[i] > 65 && InfoPay_Avg[i] < 100){
    return(1)
  }
  else return(0)
} #end I_nerdPay
```

Veteran/Armed Forces Friendly

- Proportion of population that is employed by Armed Forces: higher is better

```
V_activeVets <- function(x){ #begin V_activevets
  return(ArmedForces[x]/Pop[x]*1000)
} #end V_activevets
```

- Proportion of Veteran owned Establishments to number of Veterans: higher is better

```
V_productiveVets <- function(x){ #begin V_productiveVets
  return(VetOwned[x]/Vets[x]*10)
} #end V_productiveVets
```

Create the Algorithm

```
#Function to Calculate Pleasantness Score
Pleasant_score <- function(x){ #begin Pleasant function
    #return(t(rowSums(scores)))
    return(scores$L_rent[x] + scores$L_commute[x] + scores$L_popDensity[x] + scores$L_telework[x] + sco
}#end pleasant function
```

Run the Test!!

```
#Create a variable to hold the number of cities in the dataset
num_cities = nrow(myBestCities)
#Create a data frame to hold the scoring parameters
scores <- data.frame(L_rent=numeric(num_cities), L_commute=numeric(num_cities), L_popDensity=numeric(nu
Pleasantness=numeric(num_cities))

attach(scores)
```

```r
#Loop through each city in myBestCities to add row names and populate the scoring parameters

  for(i in 1:num_cities){ #begin loop
  row.names(scores)[i]       <- row.names(myBestCities)[i]
  scores$L_rent[i]           <- L_rent(i)
  scores$L_commute[i]        <- L_commute(i)
  scores$L_popDensity[i]     <- L_popDensity(i)
  scores$L_telework[i]       <- L_telework(i)
  scores$L_homeVal[i]        <- L_homeVal(i)
  scores$I_nerdPopDensity[i] <- I_nerdPopDensity(i)
  scores$I_nerdPay[i]        <- I_nerdPay(i)
  scores$V_activeVets[i]     <- V_activeVets(i)
  scores$V_productiveVets[i] <- V_productiveVets(i)
  scores$Pleasantness[i]     <- Pleasant_score(i)
}
#apply(scores, 1, Pleasant_score)#end loop
```

Below are all of the calculated parameters for the pleasantness score and the final score for each city:

```r
print(scores)
```

```
##                                    L_rent L_commute L_popDensity
## Madison.city..Wisconsin         1.1037528 0.5235602   0.32927231
## Minneapolis.city..Minnesota     1.1961722 0.4484305   0.14107755
## San.Francisco.city..California  0.6720430 0.3278689   0.05821027
## Austin.city..Texas              1.0224949 0.4366812   0.37690336
## Philadelphia.city..Pennsylvania 1.1198208 0.3144654   0.08787732
## New.York.city..New.York         0.8333333 0.2551020   0.03701990
## Los.Angeles.city..California    0.8510638 0.3424658   0.12357426
## Seattle.city..Washington        0.9165903 0.3937008   0.13791391
## Portland.city..Oregon           1.0928962 0.4098361   0.22856098
## Miami.city..Florida             1.0570825 0.3816794   0.08979966
## Charlottesville.city..Virginia  1.0070493 0.5747126   0.23549359
##                                 L_telework L_homeVal I_nerdPopDensity
## Madison.city..Wisconsin          0.2215180         1        0.1981913
## Minneapolis.city..Minnesota      0.2887516         1        0.2517657
## San.Francisco.city..California   0.4002931         0        0.6169752
## Austin.city..Texas               0.3958299         1        0.3163881
## Philadelphia.city..Pennsylvania  0.1184530         1        0.1469260
## New.York.city..New.York          0.1851432         0        0.2236367
## Los.Angeles.city..California     0.2666335         0        0.2070099
## Seattle.city..Washington         0.4006178         0        0.4383400
## Portland.city..Oregon            0.3993313         0        0.1800691
## Miami.city..Florida              0.1509299         1        0.1230921
## Charlottesville.city..Virginia   0.3128235         0        0.3558367
##                                 I_nerdPay V_activeVets V_productiveVets
## Madison.city..Wisconsin                 1    0.5788799        1.0965382
## Minneapolis.city..Minnesota             1    0.1359200        1.5192730
## San.Francisco.city..California          0    0.4532838        1.6741299
## Austin.city..Texas                      1    0.7047154        1.6238315
## Philadelphia.city..Pennsylvania         1    0.2044553        0.8887457
## New.York.city..New.York                 0    0.3324717        2.0354100
## Los.Angeles.city..California            1    0.2821268        2.4062924
## Seattle.city..Washington                0    2.1440542        1.5451558
## Portland.city..Oregon                   1    0.2552349        1.4756561
## Miami.city..Florida                     1    0.4255777        4.5399188
## Charlottesville.city..Virginia          1    2.1851639        3.6969001
##                                 Pleasantness
## Madison.city..Wisconsin             5.051713
## Minneapolis.city..Minnesota         4.981391
## San.Francisco.city..California      4.202804
## Austin.city..Texas                  5.876844
## Philadelphia.city..Pennsylvania     3.880744
## New.York.city..New.York             3.902117
## Los.Angeles.city..California        5.479166
## Seattle.city..Washington            5.976373
## Portland.city..Oregon               5.041585
## Miami.city..Florida                 7.768080
## Charlottesville.city..Virginia      9.367980
```

The Results are In!!

```
scores[order(scores$Pleasantness, decreasing = TRUE),10, drop = FALSE]
```

```
##                                 Pleasantness
## Charlottesville.city..Virginia      9.367980
## Miami.city..Florida                 7.768080
## Seattle.city..Washington            5.976373
## Austin.city..Texas                  5.876844
## Los.Angeles.city..California        5.479166
## Madison.city..Wisconsin             5.051713
## Portland.city..Oregon               5.041585
## Minneapolis.city..Minnesota         4.981391
## San.Francisco.city..California      4.202804
## New.York.city..New.York             3.902117
## Philadelphia.city..Pennsylvania     3.880744
```

I think this assessment seems accurrate. I am suprised that LA ranked higher than Madison due to the fact that I lowered the score for high population density and raised the score for bikeability. I was also suprised by Charlettesville, VA being my top rated city. I have never been there or heard anything about it, but after talking to some family and friends, it seemed to make sense to them, based on what they know about me.

To make the algorithm more acurate, I would tweak the Life Style bucket parameters in the formula such as commute time score and the work at home score.